

Designing a Control Card: Pin Compatibility

Thomas Volinski
3/31/11

Executive Summary:

When designing a control card it is very important that the signals going from the microcontroller to the control card pins match previously designed control cards. If they are not matched accordingly the control card will not send the correct signals to the driver board. This application note will go over the steps of selecting which MSP430 signals should be matched with which pins of the control card. This pin placement will be based off of the already designed Stellaris Cortex M3 control card.

Keywords: MSP430, Stellaris M3, control card, DRV8824

Introduction:

The MSP430 is a 16-bit microcontroller made by Texas Instruments (TI). It can be used for many different controller applications because of its very low power consumption. For this application note, the MSP430 is used to control a motor driver to drive a stepper motor. The MSP430 uses Pulse Width Modulation (PWM) to control the motor driver. These signals are the vital signal to be able to control the motor. They must be mapped correctly on the control card in order for the stepper motor to work.

The Stellaris M3 Cortex is a 32-bit microcontroller. It is much more powerful than the 16-bit MSP430. It also is used for controlling and driving a stepper motor. The Stellaris control card's pins already compatible with a custom DRV8824 EVM with a DIMM 100 slot. The control card will connected to the EVM through this slot. Then the micro controller will send signals to the DRV8824 motor driver to control the stepper motor.

Objective:

The objective of this application note is to show how the pins of a new control card with an MSP430 need to be mapped so that this card will be able to correctly communicate with the DRV8824. The mapping of these pins will be based on the already compatible Stellaris M3 Cortex.

Steps

1. Get the schematic of the Stellaris M3 Cortex, and figure out which pins the essential signals are connected.
2. Compare these signals to the MSP430 signals and find the equivalent signals on the MSP430.
3. Map the MSP430 signals to the same pin that the equivalent Stellaris M3 Cortex pins are connected too.

Step 1: Finding the Essential Pins on the Stellaris M3 Cortex

The first step is to look at the pin layout and see what useful information can be gathered from it. Figure 1 shows some interesting things. The first set of six pins on either side of the card is only used for the RS232 protocol. RS232 protocol is the way the PC will communicate with the microcontroller. This card uses a 3-pin RS232 which consists of receive, transmit and ground pins. Figure 1 shows that pins 1 and 51 are connected with 3.3 V. This is the voltage coming from the USB cable. Pin 2 is the RS232 receive pin and pin 52 is the RS232 transmit pin. Also pin 6 and 56 are connected to ground.

The second set of 16 pins on either side consists of all analog pins and a few ground pins. But, which analog pins are which? The good thing about the ARM Cortex M3 architecture is that all of the pins in a certain group are all multiplexed together so, in this case, any analog pin can be mapped to these pins.

The last section of pins contains General Purpose Input/Output (GPIO) pins, with some 5 Volt and ground pins here and there. The same goes for the GPIO pins as the analog pins, they are all multiplexed together. The very last few pins are used for the JTAG protocol. This protocol is used for debugging and has the following 4 pins TDI, TDO, TCLK, and TMS.

In conclusion, the first section of pins will be the same on the MSP430 control card. The section of pins will contain all of the analog pins from the MSP430. The final section of the MSP430 will have the JTAG pins mapped exactly like the Stellaris card. Also all of the ground and high voltage pins will be mapped exactly the same.

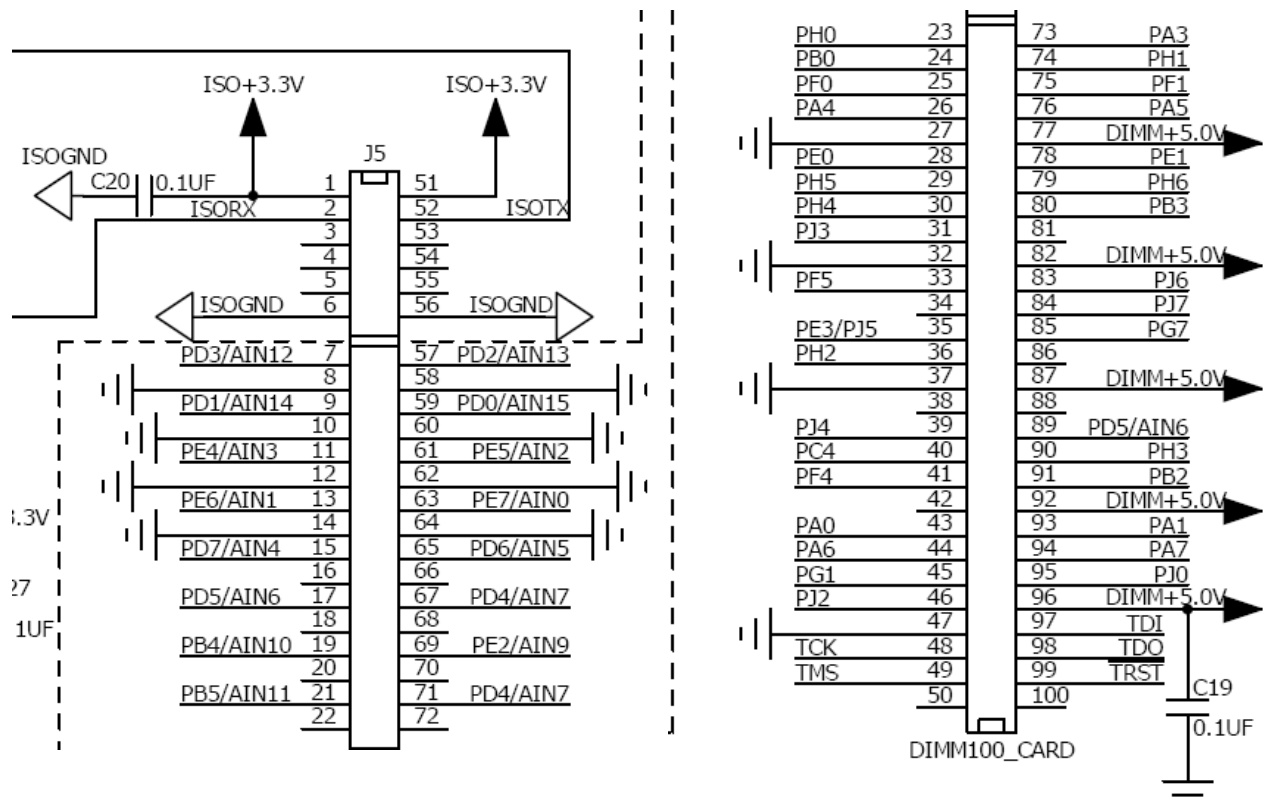


Figure 1. Pin Layout of Stellaris M3 Cortex

Step 2: Finding Important Signals on the MSP430

Looking at pieces of the MSP430 schematic in Figure 2-5, there are many pins that are either connected to nothing or connected to other components that will be on the control card. In Figure 2, it can be seen that the important pins are pin 2 through 6. These are analog pins that must be connected to the second set of pins in the control card. All of the other pins in figure 2 are either empty or are connected to other components. Figure 3 shows 16 more pins of the MSP430. The only important pin is pin 32. This pin is the receive pin of the RS232 protocol. Figure 4 shows pins 33-48 of the MSP430. Most of these pins are important. Pin 33 is the transmit pin of the RS232 protocol. Pins 37-47 are the most important pins. They are all of the control signals for the motor. These pins include the direction of the motor and the PWM signal that controls the speed of the motor. Figure 5 has some important pins as well. Pin 59-61 are analog signals and pin 54-57 are the 4 JTAG pins.

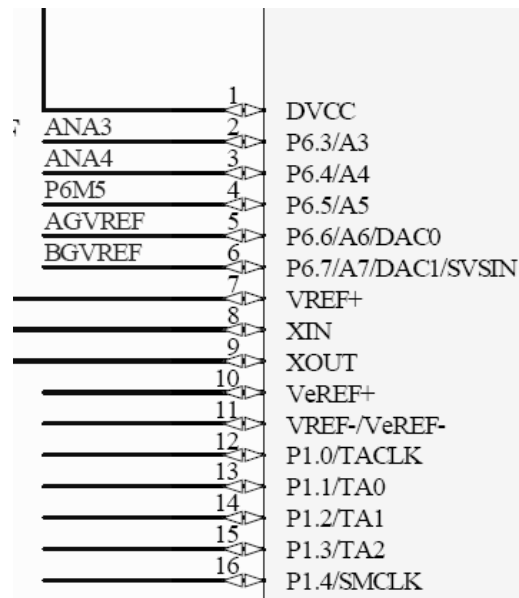


Figure 2. Pins 1-16 of the MSP430

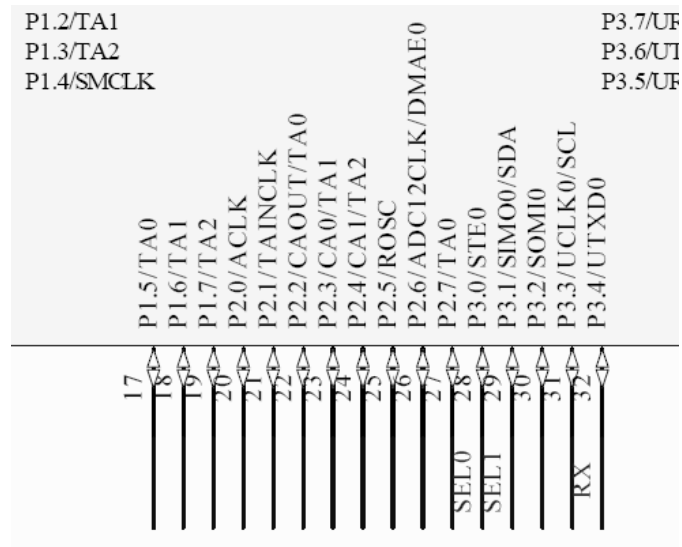


Figure 3. Pins 17-32 of the MSP430

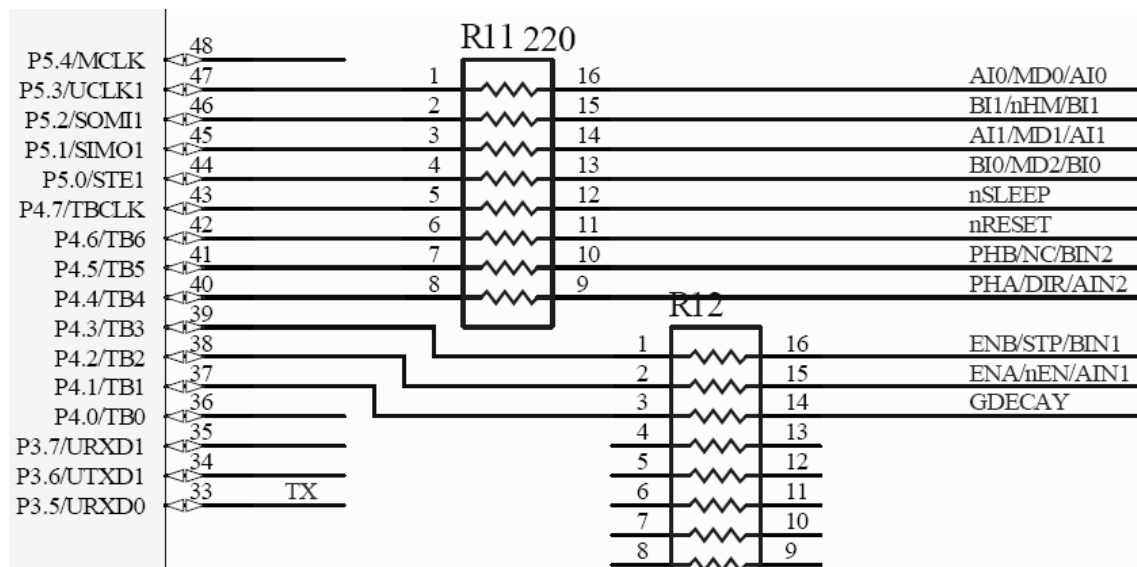


Figure 4. Pins 33-48 of the MSP430

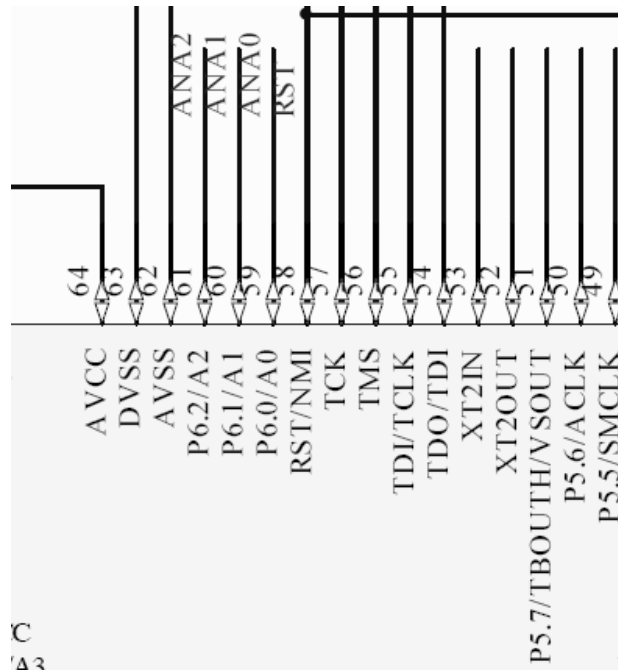


Figure 5. Pins 49-64 of the MSP430

Step 3: Mapping the MSP430 Signals to the Appropriate Control Card Pins

Now that all the pins are known it is time to map them to the appropriate pin. All ground and 5 volt pins will stay the same as the Stellaris control card. Figure 6 shows the mapping of the pins. All other pins do not have an effect on this application and can be hooked up to any pin available.

Mapping of Pins		
Control Card Pin	MSP430 Pin	
2	32	RX
52	33	TX
7	2	Analog
57	3	
9	4	
59	5	
11	6	
61	17	
13	18	
63	19	

23	37	PWM signals
73	38	
24	39	
74	40	
25	41	
75	42	
26	43	
76	44	
28	45	
78	46	
29	47	
98	54	TDO
97	55	TDI
49	56	TMS
48	57	TCLK

Figure 6. Mapping of MSP430 pins to control card pins

Conclusion:

After completing these steps one should be ready to input their control card design using a PCB design program such as PADS or Alltium Designer. Although these steps were for making a control card pin mapping for a specific microcontroller based off another specific control card, they should be easily expandable to design any control card using any microcontroller.

References:

CPG004_SCH. Texas Instruments. PDF. (DRV88xx Schematic)

CC – LM3S9B95 Schematic Rev. 0. Texas Instruments. PDF.